TIC-TAC-TOE

Proiect realizat de:

Modrea Elena-Alexandra

Grupa 30233, An3, Seria A

Profesor coordinator: Itu Razvan

1. Proictul ales. Problema pe care o rezolva.

Proiectul ales este implementarea jocului TIC-TAC-TOE utilizand Arduino si un keypad.

Am ales sa implementez acest proiect deoarece dezvolta gandirea practica a jucatorului punandu-i mintea la incercare.

Chiar daca la prima vedere TIC-TAC-TOE pare un joc usor, este important insa si adversarul deoarece poate deveni un joc greu de castigat.

1. Solutia problemei. Implementarea.

Pentru a implementa acest proiect am folosit Arduino Uno, un breadboard, un buton, o matrice de butoane 4x4, 6 fire male-female, 2 fire male-male, iar pentru a afisa jocul am folosit Processing.

Butonul este folosit pentru a se putea alege intre cele 2 variante de joc: Multi-player si Player with IA component.

Matricea de butoane este folosita pentru a simula frame-ul specific TIC-TAC-TOE, insa deoarece nu am gasit o matrice de butoane de 3x3, am folosit una de 4x4, fara a lega o linie si o coloanal la Arduino.

Pentru a lucra cu matricea mai usor, am folosit urmatoarea functiei din biblioteca Keypad.h:

**Keypad** customKeypad = **Keypad**( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

Am implementat urmatoarele functii:

Setup() – am initializat interfata seriala, precum si butonul la pinul 8.

Loop() – contine atat logica pentru a alege ce fel joc este ales astfel:

O imagine care conține text

Descriere generată automatDaca AI=true atunci jocul o sa fie setat pe Play with AIComponent

Daca AI=false jocul este setat pe Multiplayer.

Variabila ok este folosita pentru a ne asigura ca alegerea este facuta doar la inceput, fara a o putea modifica pe parcursul jocului.

De asemenea, in loop() se verifica daca avem un castigator, sau daca am ajuns la finalul jocului cu ajutorul unei functii.

printGrid() – este folosita pentru a afisa selectiile jucatorilor

number(char keyn) – este folosita pentru a da fiecarui buton un numar specific

makePlay()- contine logica jocului care este implementata dupa cum urmeaza:

O imagine care conține text

Descriere generată automat n- reprezinta numarul butonului ce a fost apasat; este actualizat vectorul ce retine alegerile jucatorilor numai daca acel buton nu a fost apasat inainte.

Vectorul se transmite catre Processing.

makePlayAIComponent() – e similara functiei anterioare, aceasta fiind folosita pentru cazul in care este ales jocul cu AI Component

checkOpponent()- este folosita pentru a verifica daca adversarul este pe cale sa castige; nu sunt toate posibiliatile trecute deoarece jocul este realizat pe modul Usor pentru a da sansa si adversarului sa castige.

movesTTT()- este folosita pentru a stii cate cellule au fost marcate de catre jucatori.

arduinoMove()- contine logica pentru mutarile lui AIComponent:

* Prima data se verifica daca colturile sunt libere, in caz afirmativ mutarea este una din colturi, random;
* Apoi este apelata functia checkOpponent(), si in cazul in care adversarul este pe cale sa castige, mutarea va fi in acel loc. In caz contrar, se va alege o celula libera in mod random.

O imagine care conține text

Descriere generată automatcheckIfWinner () - verifica daca exista castigatori astfel:

O imagine care conține text

Descriere generată automatIn Processing am create frame-ul in care vom desena jocul:

O imagine care conține text

Descriere generată automatDesenarea lui X si O a fost facuta in urmatorul fel:

Pentru a putea sti cine a castigat in Processing, am trimis prin vectorul grid, al 10-lea element ca fiind 0- nu se stie, 1- X castiga, 2- O castiga.

1. Diagrama circuitului

O imagine care conține text, electronice

Descriere generată automat

1. Bibliografie

<https://roboticsbackend.com/arduino-push-button-tutorial/>

<https://playground.arduino.cc/Code/Keypad/>

https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/